Journal of Information Security and Applications 55 (2020) 102629

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Reversible and recoverable authentication method for demosaiced images using adaptive coding technique

Xiaoyu Zhou [a], Wien Hong [b], Shaowei Weng [c], Tung-Shou Chen [b,*], Jeanne Chen [b]

[a] Dept. of Electrical and Computer Eng., Nanfang College of Sun Yat-Sen University, China
[b] Dept. of Computer Science and Info. Eng., National Taichung Univ. of Science and Tech., Taiwan
[c] School of Information Science and Engineering, FuJian University of Technology, China

ARTICLE INFO

Keywords:
Authentication
Image recovery
Image demosaicing
Reversibility

ABSTRACT

This paper proposes a reversible authentication scheme for demosaiced images with the capability to approximately recover original contents of tampered parts. The existing methods protect the marked images to a large extent; however, they cannot detect some intentional modifications, or lack the self-recovery capability of tampered regions. The proposed method uses the most significant bits (MSBs) of sampled components to generate recovery codes, and embeds them into the least significant bits (LSBs) of rebuilt components. The MSBs of rebuilt components are adjusted to reduce the embedding distortion. The authentication codes are generated by hashing the MSBs of adjusted rebuilt components, and are embedded into their LSBs. By analyzing the extracted authentication and recovery codes, the tampered regions can be localized, and the original contents can be approximately recovered. If the marked image is untampered, the original demosaiced image can be exactly reconstructed. Experimental results demonstrate that the proposed method not only provides an excellent marked image quality but also achieves a satisfactory detection and recovery results.

1. Introduction

A digital camera uses a pixel sensor array to capture the intensity of light, and quantifies the captured signal as digital values. However, the pixel sensor array only captures the light intensities but their colors are indistinguishable. To discriminate colors, a color filter array (CFA) is placed in front of the pixel sensor array to confine the capture of one particular color of the incoming light. The resultant filtered color intensity captured by the pixel sensor array is called a CFA image. A common CFA is the Bayer array [1] in which green cells are twice as many as red and blue ones. A pixel in the CFA image records only the intensity of one of the three primary colors. Therefore, the intensities of other two primary colors are ignored. To reconstruct the missing colors at each location, a process called image demosaicing [6, 16, 17] is employed to translate the CFA image into a final image in which each pixel contains full color information. The resultant final image is called the demosaiced image. Fig. 1 shows the schematic diagram of the image demosaicing process.

Nowadays, most of modern image acquisition systems obtain images using single pixel sensor array overlaid with a CFA. Because the demosaiced image preserves the most valuable information of color components of the natural scene, the protection of demosaiced images from being tampered becomes a critical issue. Currently, several approaches have been developed to identify the integrity of digital images. The image forensic approaches [19, 24] detect malicious tampering by identifying the inconsistencies of modified traces between original images and tampered ones. However, approaches of this type often suffer from relatively lower accuracy and significant computation cost. The image hashing approaches [22, 26] generate a hash sequence to represent the primary features of an image. Once primary features are altered, the generated hash sequence is very different from the original one and thus, the presence of tampering is detected. However, the hash sequence has to be appended and transmitted along with the image data in order to authenticate the image. The fragile watermarking approaches [10, 13] embed authentication information into digital images by altering the pixel values. Once the marked image is tampered, the embedded authentication information is destroyed and thus the tampered regions can be localized. Since the fragile watermarking approaches require relatively low computation cost and achieve a satisfactory performance, they have attracted a number of research to investigate the authentication methods of this type.

Early fragile watermarking methods [4, 7, 10, 13, 23, 25] focus on

* Corresponding author.
E-mail addresses: wienhong@nutc.edu.tw (W. Hong), tschen.prof@gmail.com (T.-S. Chen), jeanne@nutc.edu.tw (J. Chen).

https://doi.org/10.1016/j.jisa.2020.102629

Available online 1 November 2020
2214-2126/

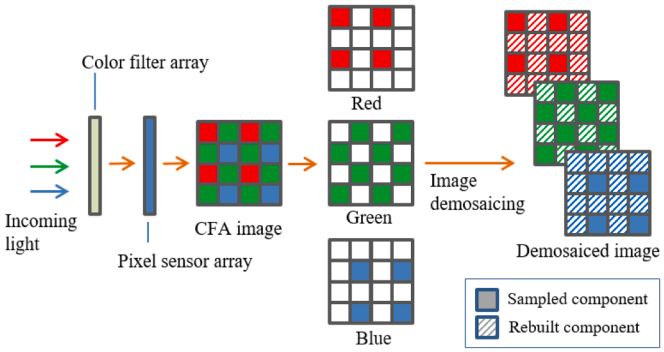


Fig.. 1. The Schematic diagram of the image demosaicing process.

detecting and locating the tampered regions. Methods of this type can be applied to images of spatial [7, 10, 23] or compressed domains [4, 13, 25]. Since images of spatial domain have richer redundancies for embedding data, authentication methods that work on this domain often provide better detection results than those of compressed ones. For a spatial domain authentication method, the original image will be inevitably distorted to produce the marked image. If the original image can be recovered from the marked image, the authentication scheme is termed reversible. While most of the existing authentication schemes are irreversible, some recent works [8, 27] have demonstrated the applicability of reversible authentication schemes.

In addition to locating the tampered regions, recent works [5, 11, 20, 21] have paid more attention to recover image contents of tampered regions. These methods not only successfully detect tampered regions but also have the capability to approximately restore the original contents of tampered parts. In 2016, Hsu and Tu [11] proposed an image detection and recovery scheme using adaptive embedding rules. Their method suggests that more recovery information of complex blocks is required in order to enhance the quality of recovered images. Therefore, they design two embedding strategies for smooth and complex blocks to improve the performance of tamper detection and recovery. In 2017, Qin et al. [21] also proposed an efficient image authentication scheme with pixel-wise recovery using overlapping embedding strategy. With the assistance of overlapping operation, the authentication and recovery information are embedded into the LSB layers. A sophisticate recovery mechanism is employed to efficiently recover the damaged bits resulting from image tampering. Qin et al. use the equation $R = \Psi \times M$ to generate recovery codes$R$, where$\Psi$and$M$ represent a random binary matrix and the information needed to be protected. Once the images are tampered, the damaged and correct parts of R and Mcould be discriminated. The embedded message M can then be recovered by simple algebraic manipulations. Overall, their method achieves an excellent detection result and provides a satisfactory recovered image quality.

While most of the spatial image authentication schemes aim on protecting grayscale images, a few studies [2, 9, 12, 14, 15, 18] have realized the prevalence of digital imaging devices and propose authentication schemes that specially designed for demosaiced images. Hu et al. [9] firstly propose an interesting reversible authentication method based on demosaiced images. Their method not only successfully detects the tampered regions, but also exactly restores the original demosaiced image if the marked image is untampered. Since the reversible constraint is imposed on their method, the traditional authentication schemes cannot be directly applied to a demosaiced image. To achieve the goal of reversible, the sampled components of a demosaiced image are intact, and the rebuilt components are utilized for embedding required information. Since the sampled components are unmodified, the original CFA image can be rebuilt, and subsequently the original demosaiced image can be reconstructed. Inspired by Hu et al.'s work, Liu et al. [14] propose an improved reversible authentication scheme for demosaiced images. The authentication information is embedded under the assistance of a 5×5 reference table and this method obtains a better marked image quality than that of Hu et al.'s. Manikandan and Masilamani [18] also propose an reversible fragile watermarking scheme for authenticating demosaicked images. However, their method provides no recoverability of the tampered regions. Belferdi et al. [2] propose an interesting fragile watermarking scheme for demosaiced images with tamper detection and restoration capability. However, their method cannot restore the original demosaiced images from untampered ones. In 2019, Hu et al. [12] also propose a tamper detection and image recovery scheme for color demosaiced images with acceptable performance. However, their method cannot restore the original demosaiced image from the marked image if the marked one is untampered.

Although Hu et al.'s [9] and Liu et al.'s [14] methods are reversible if the marked demosaiced image is untampered, their methods cannot recover the original contents of tampered regions. Moreover, the authentication information generated in their methods is irrelevant to the image features, causing some subtle modifications of the marked image undetectable. Methods [2, 12] and [18] also have the similar problems. We propose an efficient reversible authentication method for demosaiced images with recoverable capability. In our method, once the tampered regions have been detected, the original contents of tampered regions can be approximately restored. Moreover, if the marked demosaiced image is unaltered, the original demosaiced image can be perfectly reconstructed.

The rest of this paper is organized as follows. Section 2 describes the proposed method, including the authentication and recovery procedures. Section 3 presents the experimental results while the concluding remarks are given in the last section.

2. The proposed method

The proposed method is able to detect tampered regions of a marked demosaiced image and approximately recover the tampered parts. If the marked demosaiced image is untampered, the original demosaiced image can be exactly restored from its marked version. For a demosaiced image, each pixel consists of a sampled component and two rebuilt components. The proposed method only modifies the rebuilt components for data embedment and keeps the sampled components intact to ensure the restoration of the original demosaiced image. To begin with, the original demosaiced image I_D is partitioned into blocks $\{B_i\}_{i=1}^{N}$ of size 2×2, where N is the total number of blocks. The sampled and rebuilt components of the four pixels in block B_i are denoted by $\{p_{i,j}^{s}, p_{i,j}^{r_1}, p_{i,j}^{r_2}\}_{j=1}^{4}$, where the superscript s, $r_1$and r_2 are the sampled component, the first rebuilt component, and the second rebuilt component of j-th pixel of B_i, respectively. The recovery information of B_i is generated by the k-MSB of $\{p_{i,j}^{s}\}_{j=1}^{4}$. The LSBs of $\{p_{i,j}^{r_1}, p_{i,j}^{r_2}\}_{j=1}^{4}$ are used to carry data bits for authentication and recovery. The detailed procedures will be addressed in the following subsections.

2.1. The generation and embedment of recovery codes

The proposed method uses k-MSB of sampled components to generate the recovery information. Let $\varphi_i$be the concatenated k-MSB of $\{p_{i,j}^{s}\}_{j=1}^{4}$. Therefore, $\varphi_i$is a bitstream of length $4k$. By concatenating $\{\varphi_i\}_{i=1}^{N}$, we obtain a bitstream $\boldsymbol{M}$of length $4kN$. The bitstream $\boldsymbol{M}$ is then permuted using a key to obtain the permuted bitstream $\boldsymbol{M}^{\mathbf{P}}$. Segment $\boldsymbol{M}^{\mathbf{P}}$ into $4k$ equal-length sub-bitstreams, and we obtain the segmented bitstream $\{\mathbf{m}_i^{\mathbf{P}}\}_{i=1}^{N}$. To allow $\{\mathbf{m}_i^{\mathbf{P}}\}_{i=1}^{N}$can be partially recovered if some bits of them are damaged, we multiply $\mathbf{m}_i^{\mathbf{P}}$by a random binary matrix $\boldsymbol{\Psi}_i$ of size $4k \times 4k$ and then perform the modulo-2 operation. The result is a bitstream $\mathbf{r}_i$ of length $4k$. Let $\mathbf{R}$ be the concatenation of $\{\mathbf{r}_i\}_{i=1}^{N}$. Permute $\mathbf{R}$with a key $\kappa_{\mathbf{R}}$ and we obtain $\mathbf{R}^{\mathbf{P}}$. Finally, by segmenting $\mathbf{R}^{\mathbf{P}}$ into sub-bitstreams of length $4k$, we have the recovery codes $\{\mathbf{r}_i^{\mathbf{P}}\}_{i=1}^{N}$. The

procedures of generation of recovery codes are depicted in Fig. 2.

The proposed method embeds τ-bit authentication codes and 4*k*-bit recovery codes into the LSBs of rebuilt components of each block. Note that a larger τ leads to a better detectability and smaller τ results in a better image quality. To perform the embedment of block B_i, the rebuilt components $\{p_{i,j}^{r_1}, p_{i,j}^{r_2}\}_{j=1}^{4}$ are rearranged to form a 8-elememt column vector $\{p_{i,w}\}_{w=1}^{8}$. The 8×8 bitplane of $\{p_{i,w}\}_{w=1}^{8}$ can then be constructed, where the bits in ℓ-th column are the $(8-\ell+1)$-th LSB of the rebuilt components. Bits in the bitplane are numbered sequentially from 1 to 64 in a top-to-bottom and right-to-left order. The bitplane is partitioned into three areas, namely authentication code area (ACA), recovery code area (RCA), and adjustable area (AA). The ACA consists of the first through the τ-th bits in the 8×8 bitplane. Bits in ACA are used to embed the τ-bit authentication codes. The RCA consists of $(\tau+1)$-th through $(\tau+4k)$-th bits, which are used to embed the 4*k*-bit recovery codes. The remaining $(\tau+4k+1)$-th through 64-th bits belong to AA, and bits in this area can be adjusted after the embedment of recovery bits to minimize the embedding distortion. Fig. 3 is an example to illustrate of the embedment when setting $\tau=6$ and $k=5$. Notice that the proposed method uses blocks of size 2×2 as an embedding unit. Therefore, each block consists of 4 pixels. However, only two rebuilt components of each pixel are used to carry data. Therefore, 4×2=8 components can be utilized for data embedment. Once the recovery codes $\{\mathbf{r}_i^{\mathbf{P}}\}_{i=1}^{N}$ is constructed, the 4*k*-bit $\mathbf{r}_i^{\mathbf{P}}$ is embedded into the RCA of the bitplane of B_i for $1 \le i \le N$.

2.2. The modification of adjustable bits

Once the parameters τ and *k* are determined, the number of LSBs of $\{p_{i,w}\}_{w=1}^{8}$ used to embed the authentication and recovery codes is known. For example, the rearranged rebuilt components $p_{i,1}$ and $p_{i,7}$ shown in Fig. 3 use four and three LSBs to carry the codes, respectively. In the proposed method, the bits in AA and RCA will be used to generate the τ-bit authentication codes, and the generated authentication codes will be embedded into the ACA. As a result, after the bits in RCA are replaced by the recovery codes, the bits in AA can be adjusted prior to the generation of authentication codes to minimize the distortion. We propose a modification of adjustable bits (MAB) technique to perform the adjustment. Let α_w and β_w be the number bits of a rebuilt component $p_{i,w}$ used to carry α_w and β_w bits of recovery and authentication codes, respectively. Suppose $p'_{i,w}$ is the decimal value of $p_{i,w}$ where the α_w bits have been replaced by the recovery codes. Let $q_{i,w} = \lfloor p_{i,w}/2^{\beta_w} \rfloor$, $q'_{i,w} = \lfloor p'_{i,w}/2^{\beta_w} \rfloor$, and $d_{i,w} = q'_{i,w} - q_{i,w}$. We can readjust the value of $q'_{i,w}$ to minimize the distortion due to the embedment of the α_w-bit recovery codes using the equation [3]:

$$q''_{i,w} = \begin{cases} q'_{i,w} - 2^{\alpha_w}, & 2^{\alpha_w - 1} < d_{i,w} < 2^{\alpha_w} \text{ and } q'_{i,w} \ge 2^{\alpha_w}; \\ q'_{i,w} + 2^{\alpha_w}, & -2^{\alpha_w} < d_{i,w} < -2^{\alpha_w - 1} \text{ and } q'_{i,w} \le \lfloor 255/2^{\beta_w} \rfloor - 2^{\alpha_w}; \\ q'_{i,w}, & \text{otherwise.} \end{cases} \tag{1}$$

The adjusted rebuilt component $p''_{i,w}$ can then be obtained by

$$p''_{i,w} = q''_{i,w} \times 2^{\beta_w}, \ \ for \ \ 1 \le w \le 8, \tag{2}$$

Notice that the β_w-bit LSBs of $p''_{i,w}$ are all zeros. Here is an example to illustrate the adjustment technique. Suppose $p_{i,w}=80$, $\alpha_w=3$, $\beta_w=1$, and the recovery code to be placed in the fourth through second LSBs of $p_{i,w}$ is 110_2. After replacement, we have $p'_{i,w}=92$. Therefore, $q_{i,w} = \lfloor p_{i,w}/2^{\beta_w} \rfloor = \lfloor 80/2^1 \rfloor = 40$, $q'_{i,w} = \lfloor p'_{i,w}/2^{\beta_w} \rfloor = \lfloor 92/2^1 \rfloor = 46$, and $d_{i,w} = q'_{i,w} - q_{i,w} = 6$ can be obtained. Since $d_{i,w}$ is within the range $[2^{3-1}, 2^3]$ and $q'_{i,w} = 46 > 2^3$, we have $q''_{i,w} = q'_{i,w} - 2^{\alpha_w} = 46 - 2^3 = 38$. According to Eq. (2), the adjusted rebuilt component $p''_{i,w} = 38 \times 2^1 = 76$ is obtained. Note that the square error between $p'_{i,w}$ and $p_{i,w}$ is $(92-80)^2 = 144$, which is reduced to $(76-80)^2 = 16$ after adjusting $p'_{i,w}$ to $p''_{i,w}$.

2.3. The generation and embedment of authentication codes

Let $\{p''_{i,w}\}_{w=1}^{8}$ be the adjusted rebuilt components of block B_i, as described in previous section. We hash the sampled components $\{p_{i,j}^{s}\}_{j=1}^{4}$, the adjusted rebuilt components $\{p''_{i,w}\}_{w=1}^{8}$, and the position information *i* using MD5 to generate τ-bit authentication codes ac_i. The τ-bit ac_i is then embedded into the ACA of the bitplane of $\{p''_{i,w}\}_{w=1}^{8}$. Each block is processed in the same procedures, and the final marked blocks $\tilde{B}_i = \{\tilde{p}_{i,j}^{s}, \tilde{p}_{i,j}^{r_1}, \tilde{p}_{i,j}^{r_2}\}_{j=1}^{4}$ for $1 \le i \le N$ are constructed.

The decoder needs overhead data so that the authentication and recovery of the tampered image can be performed. The overhead data includes the length of authentication code τ, the length of recovery code *k*, the key $\kappa_{\mathbf{M}}$ to scramble the recovery information, the key κ_{R} to scramble the recovery code, and two predefined thresholds T_m and T_e. We use 8 bits each to record τ, *k*, T_m and T_e, and use 16 bits each to record $\kappa_{\mathbf{M}}$ and κ_{R}. Since the overhead data only requires 64 bits, it is treated as a key and can be transmitted to the receiver via a secret channel.

Here is an example to show the generation and embedment of authentication codes. Let $\{p_{i,w}\}_{w=1}^{8} = [80,78,74,73,70,65,61,56]$ be the original rebuilt components, as shown in Fig. 4(a). Suppose 01010001101010011110_2 is the recovery codes, and $\tau=6$, $k=5$. Firstly, the $\tau+1=7$-th bit through the $\tau+4k=26$-th bit in the RCA are replaced by the 20-bit recovery codes and $\{p'_{i,w}\}_{w=1}^{8}$ can be obtained, as shown in Fig. 4(b). According to Section 2.2, we adjust the bits in the AA and obtain $\{p''_{i,w}\}_{w=1}^{8}$ (Fig. 4(c)). Note that the first bit through the 6-th bit in the ACA are all zeros. Use the sampled components, adjusted rebuilt components $\{p''_{i,w}\}_{w=1}^{8}$, and the position information *i* to generate $\tau=6$-

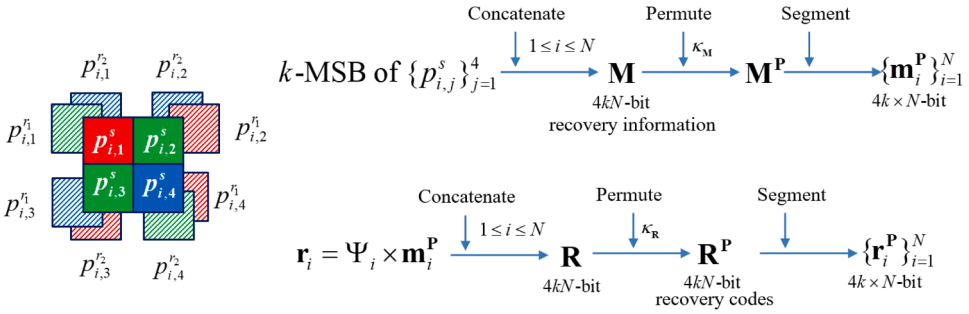


Fig.. 2. The procedures of generation of recovery codes.

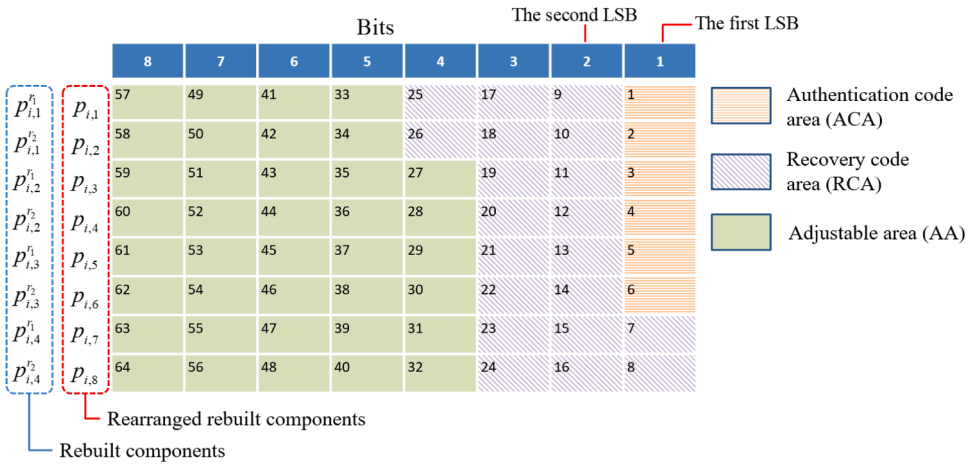


Fig.. 3. An illustration of the embedment of authentication and recovery codes.

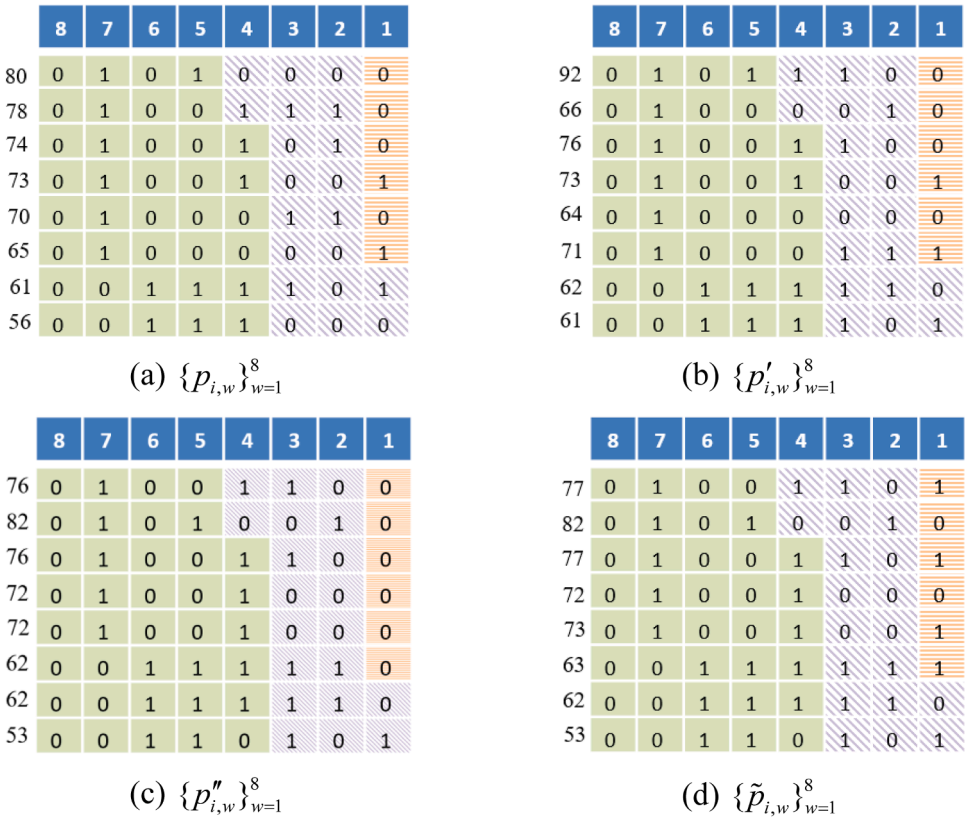


Fig.. 4. An example of generation and embedment of authentication codes.

bit authentication codes ac_i. Suppose $ac_i = 101011_2$. Replace the first bit through 6-th bit in the ACA with ac_i, and we obtain the marked rebuilt components $\{\tilde{p}_{i,w}\}_{w=1}^{8}$.

2.4. The authentication of marked demosaiced images

Let $\widehat{I}_D$be a demosaiced image to be authenticated. Firstly, we partition $\widehat{I}_D$into blocks $\{\widehat{B}_i\}_{i=1}^{N}$of size 2×2. Given the Bayer pattern, the sampled components $\{\widehat{p}_{ij}^{s}\}_{j=1}^{4}$and rebuilt components $\{\widehat{p}_{ij}^{r_1}, \widehat{p}_{ij}^{r_2}\}_{j=1}^{4}$of block $\widehat{B}_i$are known. Firstly, we arrange the bits of $\{\widehat{p}_{ij}^{r_1}, \widehat{p}_{ij}^{r_2}\}_{j=1}^{4}$into a 8×8 bitplane $\{\widehat{p}_{i,w}\}_{w=1}^{8}$, as described in Section 2.1, and then extract the τ-bit authentication codes $\widehat{e}ac_i$embedded in ACA. Because the parameter $\{\beta_{i,w}\}_{w=1}^{8}$is known, the rearranged rebuilt components $\{\widehat{p}_{i,w}\}_{w=1}^{8}$used to generate the authentication codes can be obtained as $\widehat{p}''_{i,w} = \left\lfloor \widehat{p}_{i,w} / 2^{\beta_w} \right\rfloor \times 2^{\beta_w}$. Hash $\{\widehat{p}_{ij}^{s}\}_{j=1}^{4}$, $\{\widehat{p}''_{i,w}\}_{w=1}^{8}$, and the position information $i$to obtain τ-bitauthentication codes $\widehat{a}c_i$. If $\widehat{e}ac_i = \widehat{a}c_i$, $\widehat{B}_i$is judged as untampered. Otherwise, $\widehat{B}_i$has been tampered.

A refined detection is also employed in the proposed method. Since the tampered areas of an image are likely contiguous, a block surrounded by two or more tampered blocks is likely to be also tampered. Therefore, in the refined detection stage, if the upper and lower, left and right, upper-left and lower-right, or upper-right and lower-left blocks of

an untampered block are tampered, this block is re-judged as tampered. This process is iteratively applied until no more untampered blocks are re-judged.

2.5. The recovery of demosaiced images

After detection, some blocks of the demosaiced image may be reported as tampered. With the assistance of embedded recovery codes, the tampered blocks can be approximately recovered using the scheme presented in [21]. To recover the tampered blocks, we firstly extract the recovery codes embedded in $\{\widehat{B}_i\}_{i=1}^{N}$. For each block $\widehat{B}_i$, the $4k$-bit recovery code $\hat{\mathbf{r}}_i^{\mathbf{P}}$ embedded in the RCA is extracted. By rearranging $\{\hat{\mathbf{r}}_i^{\mathbf{P}}\}_{i=1}^{N}$ into a $4kN$-bit bitstream $\hat{\mathbf{R}}^{\mathbf{P}}$, and reversely permuting $\hat{\mathbf{R}}^{\mathbf{P}}$ using the key $\kappa_{\mathbf{R}}$, we obtain the bitstream $\hat{\mathbf{R}}$ of length $4kN$. Segment and reshape $\hat{\mathbf{R}}$ into N column vector $\hat{\mathbf{r}}_i$ of length $4k$ and we obtain an $4k \times N$ matrix $\{\hat{r}_i\}_{i=1}^{N}$. Similarly for each $\widehat{B}_i$, extract k-MSB from $\{\hat{p}_{i,j}^{s}\}_{j=1}^{4}$, and the result is denoted by $\{\hat{\mathbf{m}}_i\}_{i=1}^{N}$. Concatenate $\{\hat{\mathbf{m}}_i\}_{i=1}^{N}$ and we obtain a bitstream $\widehat{\mathbf{M}}$ of length $4kN$. Permute $\widehat{\mathbf{M}}$ using the key K_M, and we have the permuted bitstream $\hat{\mathbf{M}}^{\mathbf{P}}$. By segmenting and reshaping $\hat{\mathbf{M}}^{\mathbf{P}}$ into N column vector $\hat{\mathbf{m}}_i^{\mathbf{P}}$, we obtain a matrix $\{\hat{\mathbf{m}}_i^{\mathbf{P}}\}_{i=1}^{N}$ of size $4k \times N$. Fig. 5 gives the procedures of obtaining $\{\hat{r}_i\}_{i=1}^{N}$ and $\{\hat{\mathbf{m}}_i^{\mathbf{P}}\}_{i=1}^{N}$.

Notice that if B_i is untampered, the equation

$$\hat{\mathbf{r}}_i = \boldsymbol{\Psi}_i \times \hat{\mathbf{m}}_i^{\mathbf{P}} \tag{3}$$

should be held. However, if the image $\widehat{I}_D$ is tampered, some bits in $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{m}}_i^{\mathbf{P}}$ are damaged. The positions of these damaged bits are known after the authentication process given in Section 2.4. The damaged bits in $\hat{\mathbf{m}}_i^{\mathbf{P}}$ should be recovered from the correct bits provided in $\hat{\mathbf{r}}_i$. Let $\hat{\mathbf{r}}_i^{\mathbf{C}}$ be the correct bits in $\hat{\mathbf{r}}_i$, and $\hat{\mathbf{m}}_i^{\mathbf{P,C}}$ and $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ be the correct bits and damaged bits in $\hat{\mathbf{m}}_i^{\mathbf{P}}$, respectively. If the original values of $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ are known, $\hat{\mathbf{r}}_i$ can be obtained via Eq. (3). Therefore, $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ should be the variable we have to solve. Fortunately, derived from Eq. (3), $\hat{\mathbf{r}}_i^{\mathbf{C}}$ can be written as

$$\hat{\mathbf{r}}_i^{\mathbf{C}} = \boldsymbol{\Psi}_i^{\mathbf{S}} \times \hat{\mathbf{m}}_i^{\mathbf{P}}, \tag{4}$$

where $\boldsymbol{\Psi}_i^{\mathbf{S}}$ is the sub-matrix of $\boldsymbol{\Psi}_i$ where the rows corresponding to the damaged bits in $\hat{\mathbf{r}}_i$ are taken out [21]. Because $\hat{\mathbf{m}}_i^{\mathbf{P,C}}$ and $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ are known, Eq. (4) can be rewritten as

$$\hat{\mathbf{r}}_i^{\mathbf{C}} = \boldsymbol{\Psi}_i^{\mathbf{S/C}} \times \hat{\mathbf{m}}_i^{\mathbf{P,C}} + \boldsymbol{\Psi}_i^{\mathbf{S/D}} \times \hat{\mathbf{m}}_i^{\mathbf{P,D}}, \tag{5}$$

where $\boldsymbol{\Psi}_i^{\mathbf{S/C}}$ and $\boldsymbol{\Psi}_i^{\mathbf{S/D}}$ are the sub-matrices of $\boldsymbol{\Psi}_i^{\mathbf{S}}$ where their columns correspond to the correct and damaged bits in $\hat{\mathbf{m}}_i^{\mathbf{P}}$, respectively. Since $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ is damaged, the correct value of $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ can be resolved by solving Eq. (5).

Let the resolved solution be $\hat{\mathbf{m}}_i^{\mathbf{P,R}}$. Combining $\hat{\mathbf{m}}_i^{\mathbf{P,C}}$ and $\hat{\mathbf{m}}_i^{\mathbf{P,R}}$, the correct version of $\hat{\mathbf{m}}_i^{\mathbf{P}}$, which is denoted by $\hat{\mathbf{m}}_i^{\mathbf{P,*}}$, is obtained. All the damaged bits in $\{\hat{\mathbf{m}}_i^{\mathbf{P}}\}_{i=1}^{N}$ are recovered in the same manner, and the recovered results $\{\hat{\mathbf{m}}_i^{\mathbf{P,*}}\}_{i=1}^{N}$ can be obtained. Concatenating $\{\hat{\mathbf{m}}_i^{\mathbf{P,*}}\}_{i=1}^{N}$ to form a bitstream $\widehat{\mathbf{M}}_i^{\mathbf{P,*}}$ of length $4kN$, and performing the reverse permutation using the key $\kappa_{\mathbf{M}}$, we have the corrected recovery information $\widehat{\mathbf{M}}_i^{*}$. By segmenting $\widehat{\mathbf{M}}_i^{*}$ into sub-bitstream of length $4k$, we have $\{\hat{\mathbf{m}}_i^{*}\}_{i=1}^{N}$. The kMSBs of four recovered sampled components of tampered blocks can then be obtained from $\hat{\mathbf{m}}_i^{*}$. The remaining bits of recovered sampled components are set to one except the $(8-k)$-th bit is set to zero. The procedures of obtaining $\{\hat{\mathbf{m}}_i^{*}\}_{i=1}^{N}$ are shown in Fig. 6.

In some cases, inferior solutions to Eq. (5) may be obtained due to a large-area tampering. These solutions may cause the values of recovered sampled components abnormal, resulting some noises distributed sparsely in the tampered regions. To overcome this problem, a simple refinement mechanism is proposed. Firstly, we separate the recovered CFA images into red, green and blue channels. For a component in each channel, if the absolute difference between a component and the mean value of its four neighboring components is larger than a predefined threshold T_m, the visited component is judged as abnormal. However, some components in the edges might misjudged as abnormal. Therefore, if the absolute difference between the left and right or top and bottom components of the abnormal one is larger than a predefined threshold T_e, the abnormal component is treated as a normal one. For an abnormal component, its values can be simply recovered by the mean values of its neighboring normal components.

Here is a brief example to show the procedure of solving Eq. (5). Let $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{m}}_i^{\mathbf{P}}$ be the 6×1 vectors and $\boldsymbol{\psi}_i$ be the 6×6 matrix, as shown in Fig. 7(a). Suppose the second and fourth rows of $\hat{\mathbf{r}}_i$ are damaged. Therefore, remove the second and fourth rows of $\hat{\mathbf{r}}_i$ and $\boldsymbol{\psi}_i$, and we can obtain $\hat{\mathbf{r}}_i^{\mathbf{C}}$ and $\boldsymbol{\psi}_i^{\mathbf{S}}$, as shown in Fig. 7(b). Moreover, suppose the first and fifth bits of $\hat{\mathbf{m}}_i^{\mathbf{P}}$ are also damaged. As a result, $\boldsymbol{\psi}_i^{\mathbf{S/C}}$ and $\hat{\mathbf{m}}_i^{\mathbf{P,C}}$ can be obtained by removing the elements in the first and fifth columns of $\boldsymbol{\psi}_i^{\mathbf{S}}$ and the first and fifth rows of $\hat{\mathbf{m}}_i^{\mathbf{P}}$, respectively. $\boldsymbol{\psi}_i^{\mathbf{S/D}}$ is simply the first and fifth columns of $\boldsymbol{\psi}_i^{\mathbf{S}}$. Since $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ has two bits, we may suppose the original value of $\hat{\mathbf{m}}_i^{\mathbf{P,D}}$ is $[x_1,x_2]^T$. According to Eq. (5), we obtain a linear equation with two unknowns, as shown in Fig. 7(c). With simple algebraic manipulations with modulo-2 arithmetic, $[x_1,x_2]^T = [0, 1]^T$ is obtained.

3. Experimental results

We perform several experiments in this section to demonstrate the performance and applicability of the proposed method. Nine color images of size 512×512 are used as test images, as shown in Fig. 8. The Lena, Peppers, Sailboat, Jet, House and Baboon images are obtained from SIPI image database [28], while the Fruits, Pottery and Leaves images are natural pictures taken from a digital camera.

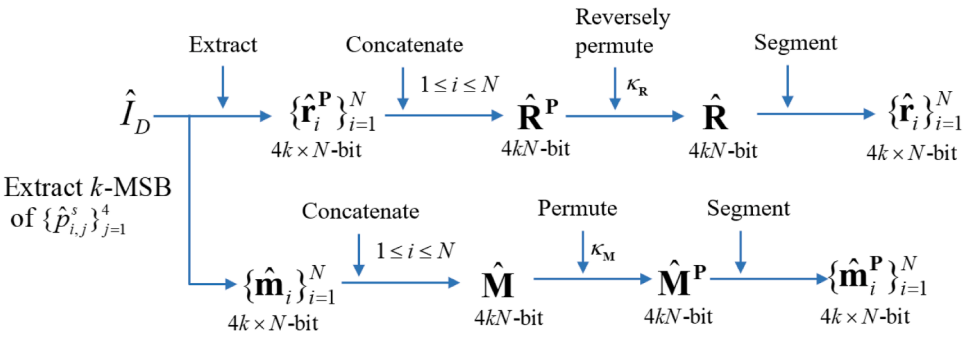


Fig.. 5. The reconstruction of recovery codes.

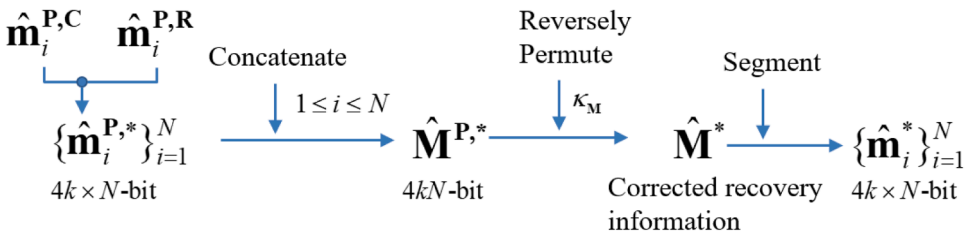


Fig.. 6. The recovery procedure of tampered blocks.

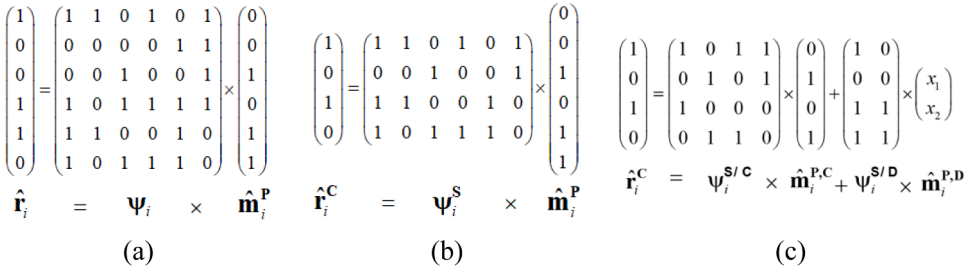


Fig.. 7. The example of solving Eq. (5).

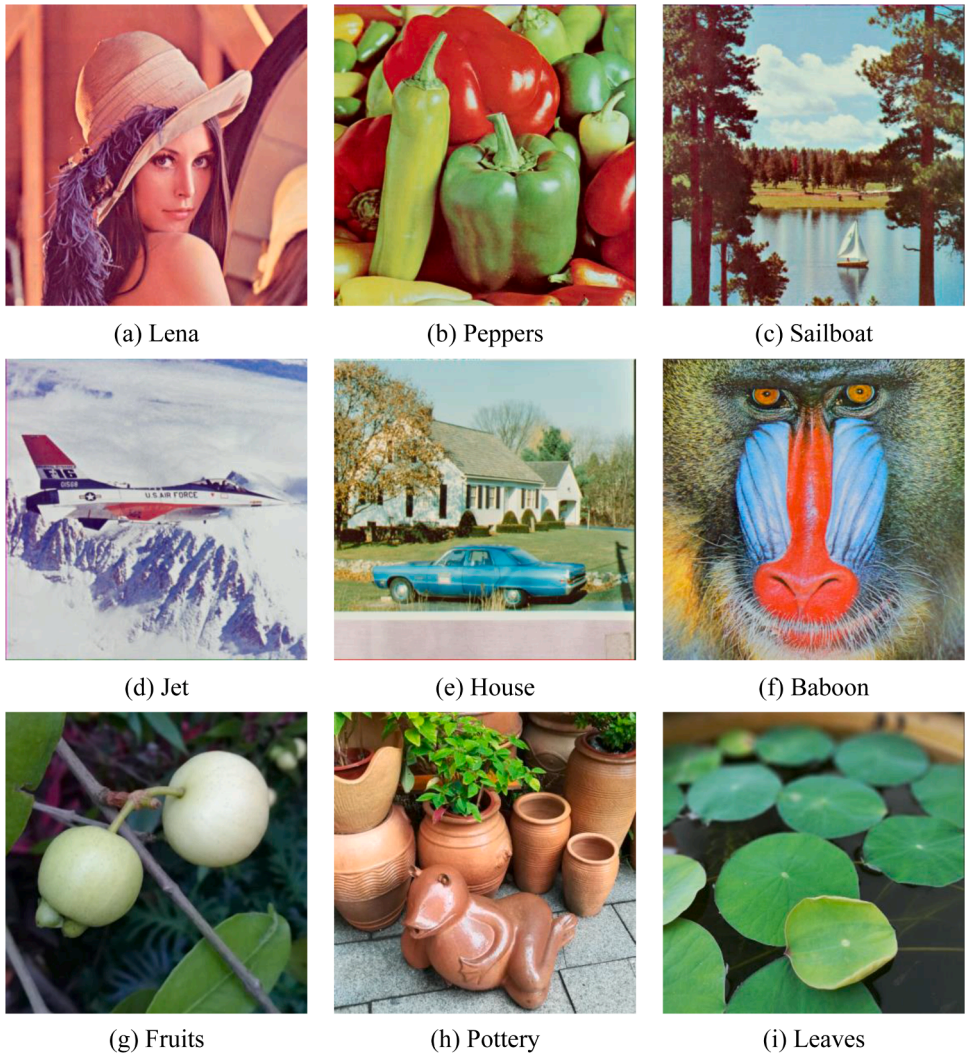


Fig.. 8. Nine test images.

We use these test images to generate the corresponding CFA images of Bayer pattern to mimic the actual ones, and use the generated CFA images to obtain demosaiced images via image demosaicing. In all experiments, a block of size 2×2 is used for embedding, authentication and recovery. We use the peak signal-to-noise ratio (PSNR, measured in dB) to evaluate the quality of the marked or recovered demosaiced image when compared with the original demosaiced image. A higher PSNR indicates that the distortion of the marked image is smaller and therefore, it has a better image quality.

Since relevant works [2, 9, 12, 14], and [18] for authentication schemes of demosaiced images have adopted different embedding strategies and have their own focuses, it is difficult to make a fair comparison of quantitative metrics between the proposed method and relevant works. As a result, we show and analyze quantitative metrics of the proposed method with different parameter values in Sections 3.1-3.3. The qualitative comparisons such as detectability, recoverability, and reversibility with other works are given in Section 3.4

3.1. Image quality comparisons

The proposed method utilizes the MAB technique shown in Section 2.2 to enhance the marked image quality. We test this technique on the nine demosaiced images with various lengths of embedded codes. We denote by $|ac|$ the length of authentication code. In the experiments, we set $|ac| = 6$ and vary the parameter k from 4 to 8. Therefore, a total of $6 + 4k$ bits are embedded into each 2×2 block. The results are shown in Table 1, where the caption 'w/o MAB' indicates that the bits in the adjustable region are not adjusted. The caption 'MAB' represents that the MAB technique is applied.

As shown in Table 1, the MAB technique effectively improves the image quality, and the improvement is more significant as the parameter k increases. For example, when $k = 4$and the MAB technique is not applied, the PSNR of the Lena image is 40.60 dB. With the application of the MAB technique, the PSNR increases to 42.67 dB. The improvement in PSNR is $42.67 - 40.60 = 2.07$ dB. In contrast, the PSNR improvement becomes $31.37 - 28.41 = 2.96$ dB when setting $k = 8$. The reason is that a larger k embeds more recovery codes, meaning that more LSBs of rebuilt components are modified. As a result, the modification of bits in the AA effectively prevents rebuilt components from being over modified.

Fig. 9 shows the image qualtiy comparison of the marked Fruits images with and without using the MAB technique. As seen in Fig. 9(a), obvious contours can be seen in the enlarged Fruits image, whereas Fig. 9(b) provides a more pleasant result.

3.2. Detectability comparisons

This section shows the detectability comparisons of the proposed method for various $|ac|$. Fig. 10(a) shows the marked Lena image. We tamper this image by placing a flower on Lena's hat (Fig. 10(b)), and the tampered region is shown in Fig. 10(c). In these experiments, $k = 5$ is set and 26,692 out of 512×512 pixels are tampered. The tampered rate is approximately 10.18%.

Fig. 11 gives the first and second stage detection results when the $|ac|$ is set to 4, 6 and 8. The detected tampered blocks are marked as black. As seen from Figs. 11(a)-(c), the rate of detected tampered blocks increases as the length of authentication code increases due to the decrease of hash collision. Figs. 11(d)-(f) give the second stage detection results. Note that a number of tampered blocks that evade the first stage detection are now detected in this stage.

During detection, most of tampered pixels will be successfully detected (true positive), and some of them will be misjudged as untampered (false negative) due to hash collisions. Similarly, while most of untampered pixels are judged as untampered (true negative), few of them are reported as tampered (false positive). Since the tampering of an image is pixel-wise and the authentication scheme of the proposed method is block-wise, a block with only one tampered pixel will be reported that all the four pixels are tampered. Moreover, the second stage detection scheme also might misjudge an untampered block as a tampered one. Therefore, the false positive situations could also happen in the proposed method. In general, the rates of false positive (FPR) and false negative (FNR) should be as small as possible. In contrast, we prefer larger rates of true positive (TPR) and true negative (TNR). Table 2 shows the detection results for different $|ac|$. As seen from this table, when only the first stage detection is applied, TPR increases as $|ac|$ increases. It is interesting to note that the FNRs are 6.55%, 1.49% and 0.45% when $|ac|$ are 4, 6 and 8, respectively. These values approximately equal the theoretical rates of hash collision ($1/2^4 \simeq 6.25\%$, $1/2^6 \simeq 1.56\%$, and $1/2^8 \simeq 0.39\%$).

The results also show that the application of the second stage detection causes a slight decrease in TNR and a slight increase in FPR; however, the improvement in TPR and FNR is significant. For example, when $|ac| = 4$, FPR slightly increases from 0.28% (first stage) to 0.39% (second stage). Nevertheless, the FNR is dramatically decreases from 6.55% to 0.06%, revealing that the application of the second stage detection effectively reduces probability of tampered pixels that are reported untampered. Although the experiments are performed on the tampering of a flower on the Lena image, experiments of other test images also show the similar trends. According to our experiments, setting $|ac| = 6$ achieves a satisfactory detection results while maintaining a high image fidelity, we would recommend to use $|ac| = 6$ in the proposed method.

3.3. The performance of recovery technique

In this section, we continue the experiments conducted in Section 3.2 to measure the influence of different length of recovery codes with respect to the quality of recovered demosaiced images. Fig. 12 shows the enlarged recovered results of the first and second stages of the Lena image for $k = 5$ and $k = 6$.

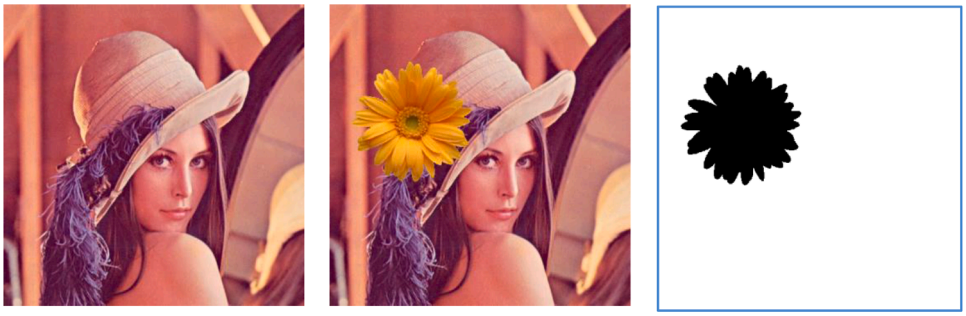
Since $k = 6$ embeds more recovery information than that of $k = 5$, it is expected more recovery information is damaged once the image has been tampered. Therefore, the damaged bits have to be recovered via Eq. (5), and the number of inferior solutions for $k = 6$ is expected more

Table. 1
PSNR comparisons with and without using the MAB technique (in dB).

	$k=4$		$k=5$		$k=6$		$k=7$		$k=8$	
images	w/o MAB	MAB	w/o MAB	MAB	w/o MAB	MAB	w/o MAB	MAB	w/o MAB	MAB
Lena	40.60	42.67	37.19	39.76	34.45	37.25	31.13	34.01	28.41	31.37
Peppers	40.56	42.55	37.17	39.54	34.45	36.99	31.14	33.70	28.45	31.02
Sailboat	40.60	42.68	37.26	39.77	34.51	37.25	31.16	34.03	28.58	31.38
Jet	40.57	42.68	37.18	39.78	34.51	37.27	31.25	34.05	28.79	31.41
House	40.57	42.67	37.18	39.76	34.42	37.25	31.15	34.06	28.59	31.38
Baboon	40.58	42.65	37.20	39.72	34.51	37.22	31.12	33.96	28.45	31.32
Friuts	40.60	42.68	37.16	39.44	34.42	36.80	31.23	33.89	28.57	31.20
Pottery	40.57	42.57	37.19	39.63	34.48	37.04	31.12	33.78	28.43	31.04
Leaves	40.62	42.69	37.23	39.78	34.47	37.26	31.19	34.05	28.60	31.37

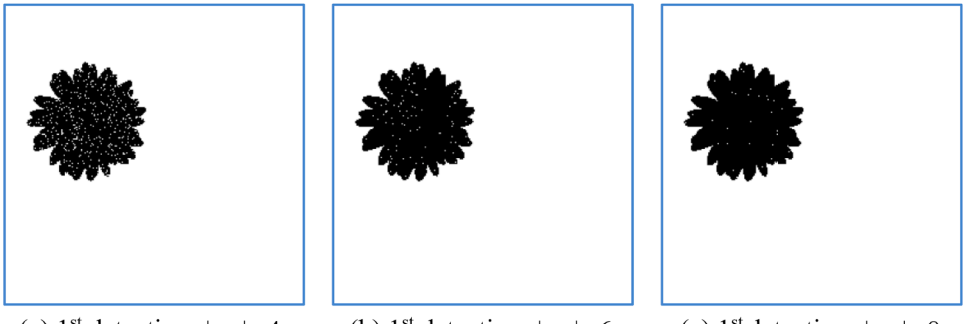
(a) Without MAB, 28.57 dB (b) With MAB, 31.20 dB

Fig.. 9. Marked images with and without using the MAB technique ($k = 8$).



(a) Marked image (b) Tampered image (c) Tampered region

Fig.. 10. The tampered Lena image.



(a) 1st detection, $|ac| = 4$ (b) 1st detection, $|ac| = 6$ (c) 1st detection, $|ac| = 8$

(d) 2nd detection, $|ac| = 4$ (e) 2nd detection, $|ac| = 6$ (f) 2nd detection , $|ac| = 8$

Fig.. 11. Results of the first and second stage detections for various $|ac|$.

Table. 2
Detection results of the Lena image for different lengths of authentication codes.

$\|ac\|$	Detection	TPR	FNR	TNR	FPR
$\|ac\|=4$	First stage	93.45%	6.55%	99.72%	0.28%
	Second stage	99.94%	0.06%	99.61%	0.39%
$\|ac\|=6$	First stage	98.51%	1.49%	99.71%	0.29%
	Second stage	99.98%	0.02%	99.59%	0.41%
$\|ac\|=8$	First stage	99.55%	0.45%	99.70%	0.30%
	Second stage	100.00%	0.00%	99.59%	0.41%

than those of $k=5$. The spots in Fig. 12 represent inferior solutions. It is quite obvious that the number of spots for $k=5$ is less than those of $k=6$, which agrees with our expectation. Nevertheless, the second recovery results for both $k=5$ and $k=6$ achieve very satisfactory results (Figs. 12(c) and (d)).

Because the recovered image quality depends on the length and the distribution of recovery codes, we vary the keys to generate the marked image, tamper the marked image by placing a flower on Lena's hat, and perform the detection and recovery for the tampered image for 100 times. The averaged PSNR of the marked and recovered images for a variety of k is shown in Fig. 13.

Fig. 13 shows that the PSNR of the recovered image at $k=4$ is the lowerest (around 44.14 dB), and significantly increases to 49.06 dB at $k=5$. However, when k is set to a larger value (6, 7, or 8), the increase of PSNR becomes insignificant. This is because a larger k provides more recovery information, which leads to a better reconstructed image quality. However, a larger k also requires to embed more bits into the RCA. Therefore, the recovery codes also have a greater chance to be damaged by image tampering, and subsequently deteriorates the recovered image quality. Fig. 13 also reveals that the quality of marked image decreases as k increases. We recommend setting $k=5$ to achieve a balance between the marked and recovered image qualities.

It is interesting to note that the PSNRs of the recovered image are higher than those of marked ones for a variety of k. The reason is that the tampered regions we recovered are sampled components, and the recovered image is constructed by demosaicing the recovered and untampered sampled components. Therefore, a majority of rebuilt components of the recovered image are identical to their original values. However, most of the rebuilt components of marked demosaiced images are modified for embedding data. Therefore, the distortion is more significant than that of the recovered one.

In the following experiments, we perform various tampering on the nine marked images and evaluate the performance of detectability and recovery. In this experiments, $|ac|=6$, $k=5$, $T_m=100$ and $T_e=30$ are set. The tampered images are shown in Figs. 14(a)-(i), and tampered rates of these nine images are 11.57%, 22.59%, 11.12%, 15.31%, 8.41%, 13.26%, 8.79%, 10.69% and 13.94%, respectively.

Fig. 15 gives the second stage detection results of the nine tampered images. The results show that the proposed method is capable of detecting various kinds of tampering. For example, the Peppers, Sailboat, Jet, Fruits and Pottery images are tampered by collaging some subimages obtained from themselves. The Lena, House and Leaves images are tampered by collaging irrelevant images obtained from the Internet. For the Baboon image, we change its nose from red to blue while

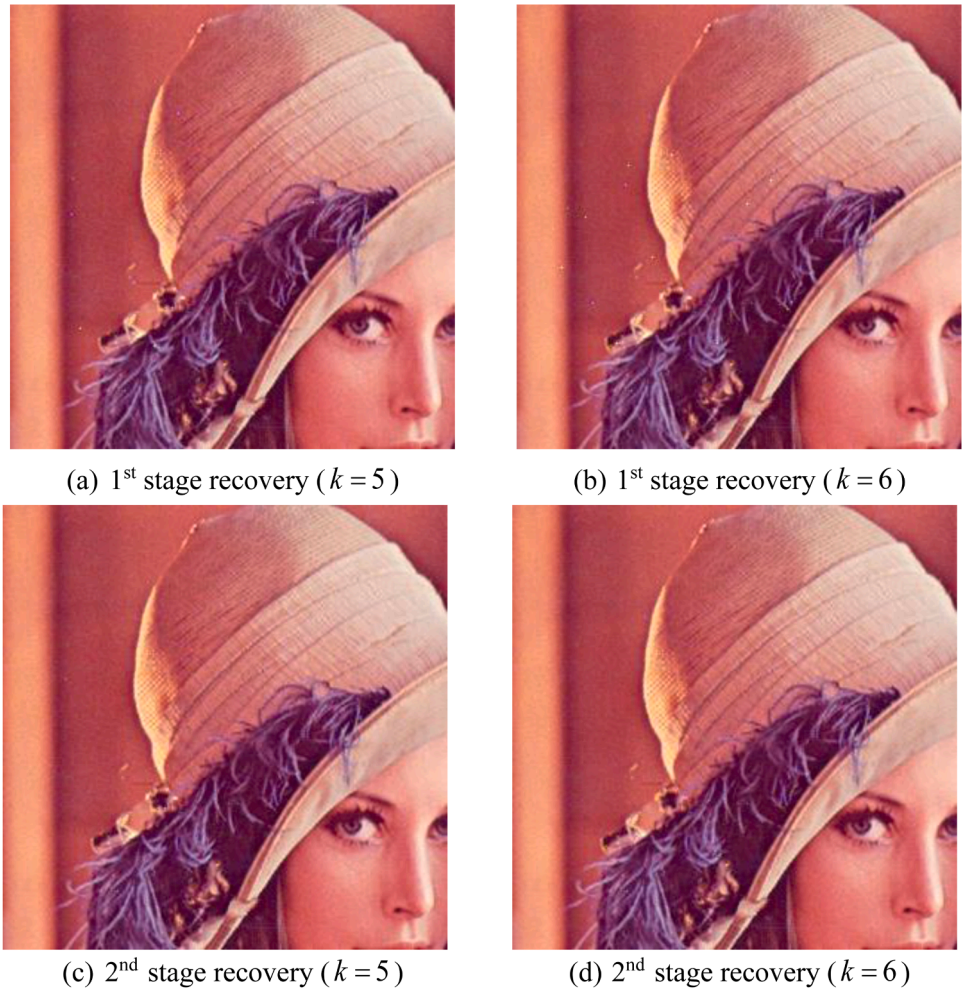


Fig.. 12. Comparisons of the recovery results for $k=5$ and $k=6$.

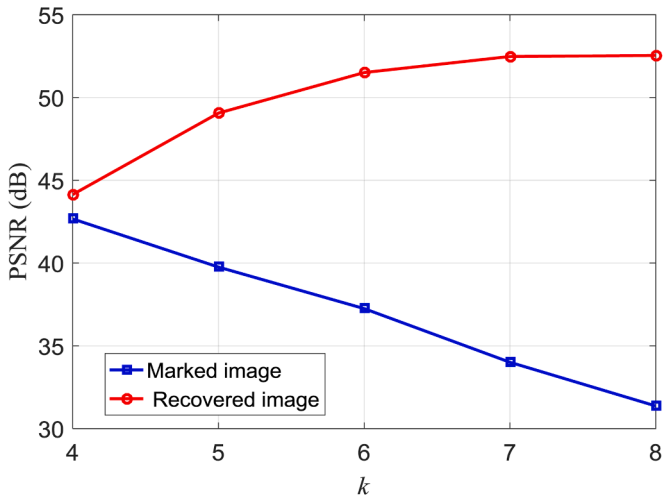


Fig.. 13. PSNR comparisons of the marked and recovered images.

preserving the original textures. As seen from Fig. 15, the proposed method can detect all these tampering with very satisfactory results.

Fig. 16 shows the second stage recovered results of the nine tampered images. As seen from these figures, the proposed method successfully recovers the tampered regions to a very satisfactory result. It is estimated that if the tampered rate is larger than 25%, few spots can still be seen in the recovered regions after the second stage detection. However, if the tampered rate is less than 25%, spots caused by inferior solutions can be almost eliminated using the second stage recovery approach.

Table 3 shows the PSNR comparisons of recovered images of the proposed method with various lengths of recovery codes (the parameter k). In this experiment, $|ac| = 6$, $T_m = 100$, and $T_e = 30$ are set. The label "Whole" used in this table indicates that the PSNR is measured by comparing the whole recovered image with the original demosaiced image, while the label "Region" only measures the PSNR of recovered tampered regions.

It can be seen from the table that as the length of recovery codes increases, the PSNRs of both recovered images and recovered tampered parts also increase. This is because the longer the recovery codes embedded, the more bits used to recover the tampered regions. Therefore, a better image quality can be achieved. It is interesting to note that for the same length of recovery codes, the PSNRs of recovered images will be higher than the PSNRs of recovered tampered regions. This is because most parts of the whole recovered images are undamaged except tampered regions. On the contrary, all pixels in the tampered regions are damaged and can only be recovered by certain length of recovery codes. Therefore, the PSNRs of the recovered tampered regions are lower than those of the whole recovered images.

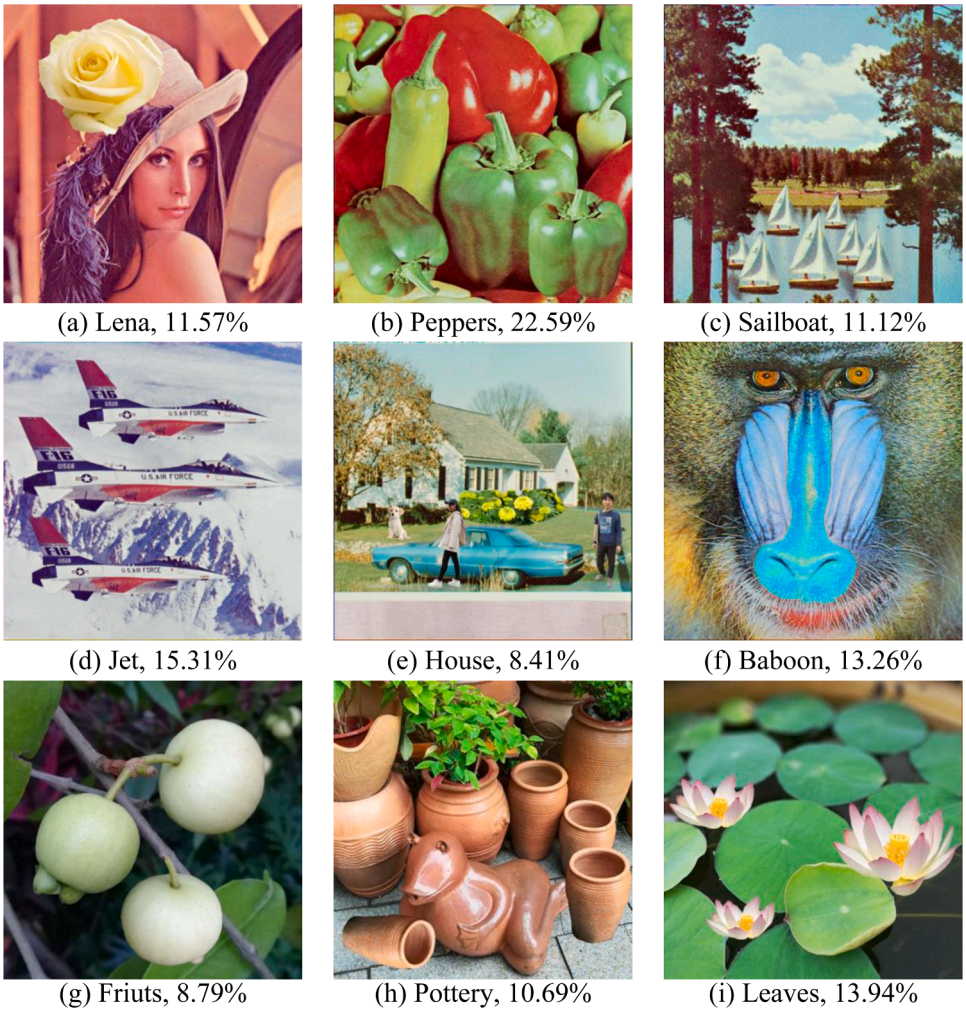


Fig.. 14. The nine tampered images.

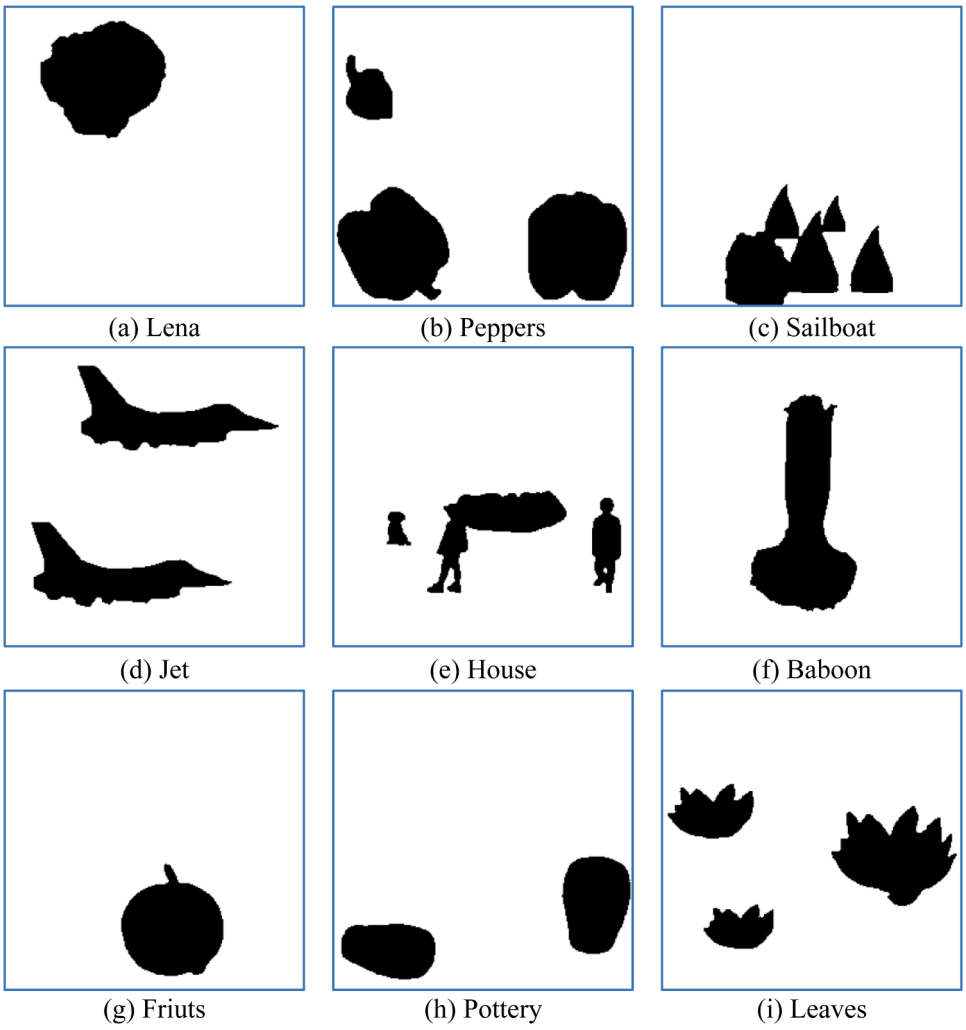


Fig.. 15. The second detection results of nine test images.

3.4. Comparisons with other works

The proposed method can approximately recover the tampered regions, and exactly reconstruct the demosaiced image if the marked image is untampered. However, to the best of our knowledge, none of the published literatures offer the aforementioned capabilities. Nevertheless, we compare the performance of proposed method with some related methods [2, 9, 12, 14, 18] in terms of detectability, recoverability and reversibility. For all the compared methods, the authentication codes are context-irrelevant random values or some properties extracted from the pixels of a demosaiced image. Therefore, some special modifications to the pixel values may not be successfully detected. Moreover, methods [9, 14] and [18] provide no capability to approximately recover the tampered regions. Besides, methods [2] and [12] are not reversible. That is, the original demosaiced cannot be restored from the untampered marked image since the sampled components are modified permanently. In contrast, the proposed method not only has excellent detectability and recoverability, but also provides the reversibility to exactly restore the original demosaiced image if the marked image is untampered. The compared results are listed in Table 4.

4. Conclusions

In this paper, we propose an efficient reversible authentication method with recovery capability based on demosaiced images. The proposed method embeds the recovery codes into the RCA, and subtly adjusts the MSBs of rebuilt components prior to the generation of authentication codes to improve the image quality. The adjusted MSBs are utilized to generate authentication codes, and the generated codes are embedded into the ACA for authentication purposes. After tamper detection, the damaged recovery codes can be recovered by solving linear equations, and the tampered blocks can be recovered via the recovered CFA image. If the marked image is untampered, the original CFA image can be obtained and subsequently the original demosaiced image can be reconstructed via the image demosaicing process. The experiments show that the proposed method not only provides an excellent marked image quality, but also offers a very satisfactory detectability, recoverability and reversibility.

Author declaration

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have

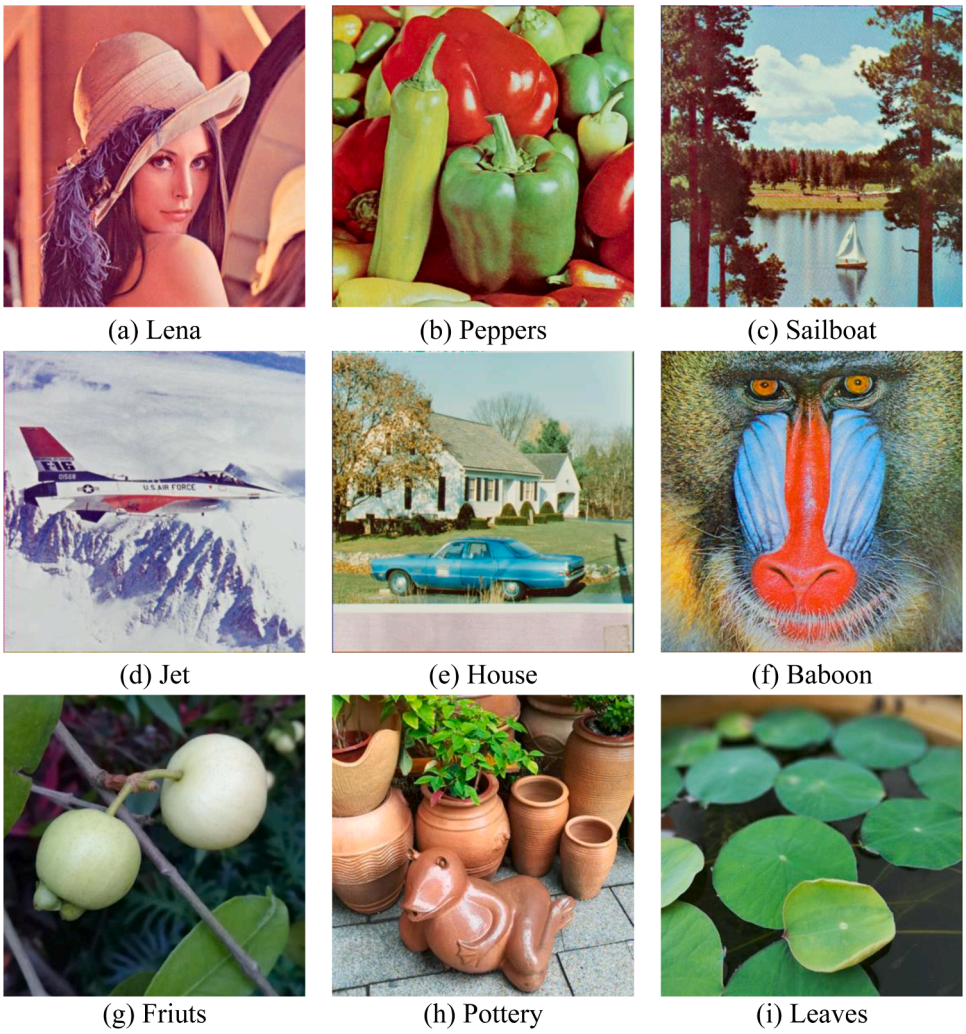


Fig.. 16. The second stage recovery results of nine test images.

Table. 3
PSNR comparisons for various lengths of recovery codes.

k	PSNR	Lena	Peppers	Sailboat	Jet	House	Baboon	Friuts	Pottery	Leaves
4	Whole	44.07	38.44	43.58	41.53	45.26	39.73	45.51	44.56	42.78
	Region	34.77	31.99	34.10	33.41	34. 60	30.96	35.04	34.91	34.26
5	Whole	50.14	42.71	49.60	47.95	50.22	41.49	50.80	49.60	48.38
	Region	40.83	36.26	40.12	39.83	39.54	32.73	40.32	39.95	39.85
6	Whole	51.73	49.91	50.55	51.61	54.01	41.62	55.23	52.61	50.84
	Region	42.39	43.46	41.04	43.50	43.32	32.86	44.76	42.94	42.30

Table. 4
The comparisons of the proposed and related methods.

Methods		Hu [9]	Liu [14]	Manikandan [18]	Belferdi [2]	Hu [12]	Proposed
Generation of authentication codes		Random values	Random values	Relationship between color channels	Averaged value of blocks	Random values	Hash function
Components to carrier of data bits		Rebuilt	Rebuilt	Rebuilt	Rebuilt and sampled	Rebuilt and sampled	Rebuilt
Detectability	Tamper the MSB of sampled components	No	No	Yes	Yes	No	Yes
	Add 8 to all components	No	Yes	No	Yes	Yes	Yes
	Scramble the components of blocks	Yes	Yes	Yes	No	Yes	Yes
Recoverability		No	No	No	Yes	Yes	Yes
Reversibility		Yes	Yes	Yes	No	No	Yes

followed the regulations of our institutions concerning intellectual property.

Declaration of Competing Interest

None.

References

[1] Bayer BE. Color imaging array. US patent 1976;3971065.
[2] Belferdi WB, Behloul A, Noui L. A bayer pattern-based fragile watermarking scheme for color image tamper detection and restoration. Multidimens Syst Signal Process 2018. https://doi.org/10.1007/s11045-018-0597-x.
[3] Chan CK, Cheng LM. Hiding data in images by simple LSB substitution. Pattern Recognit 2004;37(3):469–74.
[4] Chen TH, Chang TC. On the security of BTC-based compression image authentication scheme. Multimed Tools Appl 2018;77(10):12979–89.
[5] Chuang JC, Hu YC, Lo CC, Chen WL. Grayscale image tamper detection and recovery based on vector quantization. Int J Securi Appl 2013;7(6):209–28.
[6] Forsey A, Gungor S. Demosaicing images from colour cameras for digital image correlation. Opt Lasers Eng 2016;86:20–8.
[7] Hu YC, Chang I. Probability-based image authentication scheme for indexed color images. J Electron Imaging 2014;23(3):6–8.
[8] Hong W, Chen M, Chen TS. An efficient reversible image authentication method using improved PVO and LSB substitution techniques. Signal Process: Image Commun. 2017;58:111–22.
[9] Hu YC, Lo CC, Chen WL. Probability-based reversible image authentication scheme for image demosaicking. Future Gener Computer sys 2016;62:92–103.
[10] Hsu CS, Tu SF. Probability-based tampering detection scheme for digital images. Opt Commun 2010;283(9):1737–43.
[11] Hsu CS, Tu SF. Image tamper detection and recovery using adaptive embedding rules. Measurement 2016;88:287–96.
[12] Hu YC, Wu PJ, Chen CM, Liu YH. A novel tamper detection and image recovery technique for color image demosaicking. Recent Adv in Intell Inf Hiding and Multimed Signal Process 2019;109:217–24.
[13] Hong W, Zhou X, Lou DC, Huang X, Peng C. Detectability improved tamper detection scheme for absolute moment block truncation coding compressed images. Symmetry (Basel) 2018;10(310):1–17. https://doi.org/10.3390/sym10080318.
[14] Liu XL, Lin CC, Lin CH, Lin LJ, Qiu BJ. Reversible authentication scheme for demosaicked images without false detection. Advances in Intell Inf Hiding and Multimed Signal Process 2017;63:313–20.
[15] Lin CC, Lin CH, Liu XL, Yuan SM. Fragile watermarking-based authentication scheme for demosaicked images. In: Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing; 2015. p. 97–100.
[16] Liu Y, Wang C, Zhao H, Song J, Chen S. Bayer image demosaicking using eight-directional weights based on the gradient of color difference. Symmetry (Basel) 2018;10(6):222. https://doi.org/10.3390/sym10060222.
[17] Menon D, Calvagno G. Color image demosaicking: an overview. Signal Process: Image Commun 2011;26(8–9):518–33.
[18] Manikandan VM, Masilamani V. A context dependent fragile watermarking scheme for tamper detection from demosaicked color images. In: Proceedings of The Tenth Indian Conference on Computer Vision, Graphics and Image Processing; 2016. https://doi.org/10.1145/3009977.3009987.
[19] Piva A. An overview on image forensics. ISRN Signal Processing 2013;2013. https://doi.org/10.1155/2013/496701. Article ID 496701.
[20] Qin C, Chang CC, Chen KN. Adaptive self-recovery for tampered images based on VQ indexing and inpainting. Signal Processing 2013;93(4):933–46.
[21] Qin C, Ji P, Zhang X, Dong J, Wang J. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Processing 2017;138:280–93.
[22] Qin C, Sun M, Chang CC. Perceptual hashing for color images based on hybrid extraction of structural features. Signal Processing 2018;142:194–205.
[23] Trivedy S, Pal AK. A logistic map-based fragile watermarking scheme of digital images with tamper detection. Iran J Sci Technol. 2017;41(2):103–13.
[24] Taimori A, Razzazi F, Behrad A, Ahmadi A, Babaie-Zadeh M. A novel forensic image analysis tool for discovering double JPEG compression clues. Multimed Tools Appl 2017;vol.76(6):7749–83.
[25] Wu CM, Hu YC, Liu KY, Chuang JC. A novel active image authentication scheme for block truncation coding. Int J Signal Process, Image Process Pattern Recognit 2014;7(5):13–26.
[26] Swaminathan A, Mao Y, Wu M. Robust and secure image hashing. IEEE Trans Inf Forensics and Secur 2006;1(2):215–30.
[27] Yin Z, Niu X, Zhou Z, Tang J, Luo B. Improved reversible image authentication scheme. Cognit Comput 2016;8(5):890–9.
[28] The USC-SIPI image database. Available: http://sipi.usc.edu/database.